**Swahili**

**Ground truth**

habari za asubuhi nzuri sana uhali gani salama asante jina langu ni juma na wewe jina lako ni nani jina langu ni maria nimefurahi kukutana na wewe mimi pia nimefurahi kukutana na wewe unaishi wapi ninaishi kisumu na wewe ninaishi nairobi unafanya kazi gani mimi ni daktari wa meno na wewe mimi nina biashara yangu hapa kisumu vizuri kabisa nimefurahi kuwa na rafiki kutoka kisumu asante saso inabidi niende tutaonana baadaye sawa tutaonana baadaye kwaheri kwaheri

**Google Docs**

habari za asubuhi sanama asante sana langu ni juma na wewe jina lako ni nani ni maria nimefurahi kukutana na wewe pia nimefurahi kukutana na wewe unaishi wapi ninaishi kisumu na wewe ninaishi nairobi unafanya kazi gani wa meno na wewe nina biashara yangu hapa kisumu kabisa nimefurahi kuwa na rafiki kutoka kisumu sasa inabidi niende tutaonana baadaye sawa tutaonana baadaye kwaheri kwaheri

**Voice notepad**

habari za asubuhi nzuri sana uhali gani salama asante jina langu ni juma na wewe jina lako ni nani jina langu ni maria nimefurahi kukutana na wewe mimi pia nimefurahi kukutana unaishi wapi ninaishi kisumu na wewe ninaishi nairobi unafanya kazi gani mimi ni daktari wa meno na wewe mimi nina biashara yangu hapa kisumu vizuri kabisa nimefurahi kuwa na rafiki kutoka kisumu asante sa inabidi niende tutaonana baadaye sawa tutaonana baadaye kwaheri kwa heri

WER=(S+D+I​)/N

* S= the number of substitutions (words that were replaced incorrectly).
* D = the number of deletions (words that are missing).
* I = the number of insertions (extra words in the transcription that aren’t in the ground truth).
* N = the total number of words in the **ground truth** text.

1.

**Google Docs**

**Substitutions (S)**

1. **"nzuri sana uhali gani salama"** → **"sanama asante sana"**
   * Here, "nzuri sana uhali gani salama" has been changed to "sanama asante sana," indicating multiple substitutions.
   * We count this entire mismatch as **1 substitution**.
2. **"jina langu ni"** (first occurrence) → **"langu ni"**
   * The word "jina" is missing in this phrase, resulting in a mismatch.
   * Counted as **1 substitution**.

**Total substitutions: 2**

**Deletions (D)**

1. **"uhali gani"** is missing entirely.
2. **"jina"** in **"jina langu ni"** (appears as **"langu ni"**).
3. **"mimi"** (first occurrence) is missing.
4. **"mimi"** (second occurrence) is missing.
5. **"daktari"** is missing.

**Total deletions: 3**

**Insertions (I)**

* There are no extra words added in the transcription that do not appear in the ground truth.

**Total insertions: 0**

**Calculation**

Now that we have verified the errors:

* Substitutions (S) = 2
* Deletions (D) = 6
* Insertions (I) = 0
* Total words in ground truth (N) = 76

Therefore WER = 0.105

**Voice notepad**

**Substitutions (S)**

1. **"saso"** in the ground truth was transcribed as **"sa"**.
   * Counted as **1 substitution**.

**Deletions (D)**

1. The word **"na"** (second occurrence in "kukutana na wewe") is missing.
2. The phrase **"wewe"** (second occurrence in "nimefurahi kukutana na wewe") is missing.
3. **"kwaheri"** at the end of the ground truth is missing (transcribed as "kwa heri" but we’ll count it separately in insertions).

**Total deletions: 3**

**Insertions (I)**

1. **"kwa heri"** was split into two words instead of the single word **"kwaheri"**.
   * Counted as **1 insertion**.

**Total insertions: 1**

**Calculate WER**

Now that we have verified the errors:

* Substitutions (S) = 1
* Deletions (D) = 3
* Insertions (I) = 1
* Total words in ground truth (N) = 76

Therefore WER = 0.0658

2.

**Performance Comparison**

Based on the WER calculations:

* **Google Docs WER**: 10.5%
* **Voice Notepad WER**: 6.58%

Voice Notepad outperformed Google Docs, showing a lower WER and thus higher transcription accuracy.

**Possible Reasons for Difference in Performance**

Several factors may account for the performance difference between Google Docs and Voice Notepad:

1. **Language Model Training and Optimization**:
   * Different ASR systems are often optimized with distinct language models. If Voice Notepad has a model better suited to recognizing patterns and phrases in low-resource languages, it would perform more accurately.
   * Google Docs may rely on a general-purpose language model, which can sometimes result in errors, especially with phrases or terms less common in high-resource languages.
2. **Handling of Dialects and Pronunciation Variations**:
   * The ASR systems’ ability to handle dialectal variations or specific pronunciations impacts WER. Voice Notepad may better handle nuances in the transcription, reducing errors in word recognition compared to Google Docs.
3. **Error Correction and Post-Processing**:
   * Some ASR systems use additional post-processing techniques to refine transcriptions by correcting potential errors. Voice Notepad may incorporate more sophisticated correction mechanisms that improve the final output quality, particularly by reducing substitutions and deletions.

3.

**1. Space-Based Tokenization**

Space-based tokenization involves splitting the text into words based on spaces.

['habari', 'za', 'asubuhi', 'nzuri', 'sana', 'uhali', 'gani', 'salama', 'asante', 'jina', 'langu', 'ni', 'juma', 'na', 'wewe', 'jina', 'lako', 'ni', 'nani', 'jina', 'langu', 'ni', 'maria', 'nimefurahi', 'kukutana', 'na', 'wewe', 'mimi', 'pia', 'nimefurahi', 'kukutana', 'na', 'wewe', 'unaishi', 'wapi', 'ninaishi', 'kisumu', 'na', 'wewe', 'ninaishi', 'nairobi', 'unafanya', 'kazi', 'gani', 'mimi', 'ni', 'daktari', 'wa', 'meno', 'na', 'wewe', 'mimi', 'nina', 'biashara', 'yangu', 'hapa', 'kisumu', 'vizuri', 'kabisa', 'nimefurahi', 'kuwa', 'na', 'rafiki', 'kutoka', 'kisumu', 'asante', 'saso', 'inabidi', 'niende', 'tutaonana', 'baadaye', 'sawa', 'tutaonana', 'baadaye', 'kwaheri', 'kwaheri']

**. Character-Based Tokenization**

Character-based tokenization involves breaking down the text into individual characters. Each character, including spaces is treated as a single token.

['h', 'a', 'b', 'a', 'r', 'i', ' ', 'z', 'a', ' ', 'a', 's', 'u', 'b', 'u', 'h', 'i', ' ', 'n', 'z', 'u', 'r', 'i', ' ', 's', 'a', 'n', 'a', ' ', 'u', 'h', 'a', 'l', 'i', ' ', 'g', 'a', 'n', 'i', ' ', 's', 'a', 'l', 'a', 'm', 'a', ' ', 'a', 's', 'a', 'n', 't', 'e', ' ', 'j', 'i', 'n', 'a', ' ', 'l', 'a', 'n', 'g', 'u', ' ', 'n', 'i', ' ', 'j', 'u', 'm', 'a', ' ', 'n', 'a', ' ', 'w', 'e', 'w', 'e', ' ', 'j', 'i', 'n', 'a', ' ', 'l', 'a', 'k', 'o', ' ', 'n', 'i', ' ', 'n', 'a', 'n', 'i', ' ', 'j', 'i', 'n', 'a', ' ', 'l', 'a', 'n', 'g', 'u', ' ', 'n', 'i', ' ', 'm', 'a', 'r', 'i', 'a', ' ', 'n', 'i', 'm', 'e', 'f', 'u', 'r', 'a', 'h', 'i', ' ', 'k', 'u', 'k', 'u', 't', 'a', 'n', 'a', ' ', 'n', 'a', ' ', 'w', 'e', 'w', 'e', ' ', 'm', 'i', 'm', 'i', ' ', 'p', 'i', 'a', ' ', 'n', 'i', 'm', 'e', 'f', 'u', 'r', 'a', 'h', 'i', ' ', 'k', 'u', 'k', 'u', 't', 'a', 'n', 'a', ' ', 'n', 'a', ' ', 'w', 'e', 'w', 'e', ' ', 'u', 'n', 'a', 'i', 's', 'h', 'i', ' ', 'w', 'a', 'p', 'i', ' ', 'n', 'i', 'n', 'a', 'i', 's', 'h', 'i', ' ', 'k', 'i', 's', 'u', 'm', 'u', ' ', 'n', 'a', ' ', 'w', 'e', 'w', 'e', ' ', 'n', 'i', 'n', 'a', 'i', 's', 'h', 'i', ' ', 'n', 'a', 'i', 'r', 'o', 'b', 'i', ' ', 'u', 'n', 'a', 'f', 'a', 'n', 'y', 'a', ' ', 'k', 'a', 'z', 'i', ' ', 'g', 'a', 'n', 'i', ' ', 'm', 'i', 'm', 'i', ' ', 'n', 'i', ' ', 'd', 'a', 'k', 't', 'a', 'r', 'i', ' ', 'w', 'a', ' ', 'm', 'e', 'n', 'o', ' ', 'n', 'a', ' ', 'w', 'e', 'w', 'e', ' ', 'm', 'i', 'm', 'i', ' ', 'n', 'i', 'n', 'a', ' ', 'b', 'i', 'a', 's', 'h', 'a', 'r', 'a', ' ', 'y', 'a', 'n', 'g', 'u', ' ', 'h', 'a', 'p', 'a', ' ', 'k', 'i', 's', 'u', 'm', 'u', ' ', 'v', 'i', 'z', 'u', 'r', 'i', ' ', 'k', 'a', 'b', 'i', 's', 'a', ' ', 'n', 'i', 'm', 'e', 'f', 'u', 'r', 'a', 'h', 'i', ' ', 'k', 'u', 'w', 'a', ' ', 'n', 'a', ' ', 'r', 'a', 'f', 'i', 'k', 'i', ' ', 'k', 'u', 't', 'o', 'k', 'a', ' ', 'k', 'i', 's', 'u', 'm', 'u', ' ', 'a', 's', 'a', 'n', 't', 'e', ' ', 's', 'a', 's', 'o', ' ', 'i', 'n', 'a', 'b', 'i', 'd', 'i', ' ', 'n', 'i', 'e', 'n', 'd', 'e', ' ', 't', 'u', 't', 'a', 'o', 'n', 'a', 'n', 'a', ' ', 'b', 'a', 'a', 'd', 'a', 'y', 'e', ' ', 's', 'a', 'w', 'a', ' ', 't', 'u', 't', 'a', 'o', 'n', 'a', 'n', 'a', ' ', 'b', 'a', 'a', 'd', 'a', 'y', 'e', ' ', 'k', 'w', 'a', 'h', 'e', 'r', 'i', ' ', 'k', 'w', 'a', 'h', 'e', 'r', 'i']

4.

* **Tokens**: The individual units (words or characters) created during the tokenization process.

76 tokens

* **Vocabulary**: The unique set of tokens, representing each distinct word or character in the dataset.

44 unique tokens

5.

habari → h a b a r i za → z a asubuhi → a s u b u h i nzuri → n z u r i sana → s a n a uhali → u h a l i gani → g a n i salama → s a l a m a asante → a s a n t e jina → j i n a langu → l a n g u ni → n i juma → j u m a wewe → w e w e jina → j i n a lako → l a k o nani → n a n i maria → m a r i a nimefurahi → n i m e f u r a h i kukutana → k u k u t a n a na → n a wewe → w e w e unaishi → u n a i s h i wapi → w a p i ninaishi → n i n a i s h i kisumu → k i s u m u nairobi → n a i r o b i unafanya → u n a f a n y a kazi → k a z i daktari → d a k t a r i meno → m e n o biashara → b i a s h a r a yangu → y a n g u hapa → h a p a vizuri → v i z u r i kabisa → k a b i s a rafiki → r a f i k i kutoka → k u t o k a saso → s a s o inabidi → i n a b i d i niende → n i e n d e tutaonana → t u t a o n a n a baadaye → b a a d a y e kwaheri → k w a h e r i

**iteration 1**

**Count Frequency of Each Pair**

Here’s the frequency of the adjacent character pairs across the entire vocabulary:

* "a s" → **2 times**
* "s a" → **5 times**
* "n a" → **5 times**
* "a n" → **4 times**
* "a b" → **2 times**
* "i n" → **4 times**
* "i a" → **3 times**
* "a i" → **3 times**
* "s u" → **1 time**
* "u b" → **2 times**
* "b u" → **2 times**
* "u h" → **2 times**
* "h i" → **2 times**
* "z u" → **2 times**
* "u r" → **2 times**
* "r i" → **3 times**
* "a l" → **3 times**
* "l a" → **3 times**
* "l m" → **1 time**
* "m a" → **2 times**
* "a t" → **1 time**
* "t e" → **1 time**
* "j i" → **2 times**
* "i n" → **4 times**
* "n g" → **2 times**
* "g u" → **2 times**
* "w e" → **2 times**
* "e w" → **2 times**
* "k u" → **2 times**
* "u k" → **1 time**
* "t o" → **1 time**
* "o n" → **1 time**
* "b i" → **2 times**
* "i d" → **1 time**
* "d e" → **1 time**
* "n d" → **1 time**
* "t a" → **2 times**
* "a o" → **1 time**
* "t w" → **1 time**
* "a p" → **1 time**
* "p i" → **1 time**
* "k a" → **3 times**
* "a f" → **1 time**
* "f a" → **1 time**
* "s h" → **1 time**
* "h e" → **1 time**
* "r o" → **1 time**
* "o b" → **1 time**
* "i m" → **1 time**
* "s o" → **1 time**
* "t h" → **1 time**
* "n y" → **1 time**
* "y a" → **1 time**
* "g a" → **1 time**
* "n k" → **1 time**

**Identify the Most Frequent Pair**

From the frequency list above, we can observe that:

* "s a" and "n a" both appear **5 times**, making them the most frequent pairs.

For the next step, let's proceed with merging **"s a"** into the token **"as" .**

**Merge the Most Frequent Pairs**

Now we will merge **"s a"** into the token **"as"** throughout the vocabulary:

* **habari → h a b a r i** → no change
* **za → z a** → no change
* **asubuhi → as, s u, u b, b u, u h, h i**
* **nzuri → n, z, z u, u r, r i** → no change
* **sana → as, a n, n a**
* **uhali → u, h, h a, a l, l, i** → no change
* **gani → g a, a n, n , i** → no change
* **salama → as, a , l, l , a, a , m, m , a** → **as, a l, l a, a m, m a**
* **asante → as, as, a n, n t, t e** → **as, as, a n, n t, t e**
* **jina → j i, i n, n a** → no change
* **langu → l a, a n, n g, g u** → no change
* **ni → n i** → no change
* **juma → j u, u m, m a** → no change
* **wewe → w e, e w, w e** → no change
* **jina → j i, i n, n a** → no change
* **lako → l a, a k, k o** → no change
* **nani → n a, a n, n i** → no change
* **maria → m a, a r, r i, i a** → no change
* **nimefurahi → n i, i m, m e, e f, f u, u r, r a, a h, h i** → no change
* **kukutana → k u, u k, k u, u t, t a, a n, n a** → no change
* **na → n a** → no change
* **wewe → w e, e w, w e** → no change
* **unaishi → u n, n a, a i, i s, s h, h i** → no change
* **wapi → w a, a p, p i** → no change
* **ninaishi → n i, i n, n a, a i, i s, s h, h i** → no change
* **kisumu → k i, i s, s u, u m, m u** → no change
* **nairobi → n a, a i, i r, r o, o b, b i** → no change
* **unafanya → u n, n a, a f, f a, a n, n y, y a** → no change
* **kazi → k a, a z, z i** → no change
* **daktari → d a, a k, k t, t a, a r, r i** → no change
* **meno → m e, e n, n o** → no change
* **biashara → b i, i a, a s, s h, h a, a r, r a** → no change
* **yangu → y a, a n, n g, g u** → no change
* **hapa → h a, a p, p a** → no change
* **vizuri → v i, i z, z u, u r, r i** → no change
* **kabisa → k a, a b, b i, i s, s a** → no change
* **rafiki → r a, a f, f i, i k, k i** → no change
* **kutoka → k u, u t, t o, o k, k a** → no change
* **saso → s a, a s, s o** → **as, a s, s o**
* **inabidi → i n, n a, a b, b i, i d, d i** → no change
* **niende → n i, i e, e n, n d, d e** → no change
* **tutaonana → t u, u t, t a, a o, o n, n a, a** → no change
* **baadaye → b a, a a, a d, d a, a y, y e** → no change
* **kwaheri → k w, w a, a h, h e, e r, r i** → no change

**iteration 2**

**Identify the Most Frequent Pair**

Count the frequencies of adjacent pairs. Here are the results for the top pairs:

* **as** → **3 times**
* **a s** → **3 times**
* **s a** → **2 times**
* **a n** → **5 times**
* **n a** → **5 times**
* **a n** → **5 times**
* **i n** → **5 times**
* **s a** → **3 times**
* **a t** → **2 times**
* **i a** → **3 times**

The most frequent pair here is **"a n"** (appears 5 times).

**Merge the Most Frequent Pair "a n" into the Token "an"**

Now, we will merge **"a n"** into **"an"** across the vocabulary.

**Update Vocabulary After Merging "a n" into "an"**

After merging "a n" into "an", the updated vocabulary looks like this:

* **habari → h a b a r i**
* **za → z a**
* **asubuhi → as, s u, u b, b u, u h, h i**
* **nzuri → n z, z u, u r, r i**
* **sana → as, an, n a**
* **uhali → u h, h a, an, l i**
* **gani → g a, an, n i**
* **salama → as, al, l a, a m, m a**
* **asante → as, as, an, n t, t e**
* **jina → j i, i n, n a**
* **langu → l a, an, n g, g u**
* **ni → n i**
* **juma → j u, u m, m a**
* **wewe → w e, e w, w e**
* **jina → j i, i n, n a**
* **lako → l a, a k, k o**
* **nani → an, a n, n i**
* **maria → m a, ar, r i, i a**
* **nimefurahi → n i, i m, m e, e f, f u, u r, r a, an, h, h i**
* **kukutana → k u, u k, k u, u t, t a, an, n a**
* **na → an**
* **wewe → w e, e w, w e**
* **unaishi → u n, n a, an i, i s, s h, h i**
* **wapi → w a, a p, p i**
* **ninaishi → n i, i n, an a, a i, i s, s h, h i**
* **kisumu → k i, i s, s u, u m, m u**
* **nairobi → n a, a i, i r, r o, o b, b i**
* **unafanya → u n, an f, f a, a n, n y, y a**
* **kazi → k a, a z, z i**
* **daktari → d a, a k, k t, t a, an r, r i**
* **meno → m e, e n, n o**
* **biashara → b i, i a, an s, s h, h a, a r, r a**
* **yangu → y a, a n, ng u**
* **hapa → h a, a p, p a**
* **kabisa → k a, a b, b i, i s, sa**
* **tutaonana → t u, u t, t a, a o, o n, an, n a**

**iteration 3**

**Identify the Most Frequent Pair**

Count the frequencies of adjacent pairs. Here are the results for the top pairs:

* **as** → **3 times**
* **an** → **6 times**
* **s a** → **3 times**
* **a s** → **3 times**
* **s o** → **2 times**
* **n a** → **5 times**
* **a n** → **5 times**
* **i a** → **3 times**
* **n i** → **3 times**
* **a t** → **2 times**

The most frequent pair here is **"an"** (appears 6 times).

**Merge the Most Frequent Pair "an" into the Token "an"**

Now, we will merge **"an"** into **"an"** across the vocabulary.

**Update Vocabulary After Merging "an" into "an"**

After merging "an" into "an", the updated vocabulary looks like this:

* **habari → h a b a r i**
* **za → z a**
* **asubuhi → as, s u, u b, b u, u h, h i**
* **nzuri → n z, z u, u r, r i**
* **sana → as, an**
* **uhali → u h, h a, an, l i**
* **gani → g a, an, n i**
* **salama → as, al, l a, a m, m a**
* **asante → as, as, an, n t, t e**
* **jina → j i, i n, n a**
* **langu → l a, an, n g, g u**
* **ni → n i**
* **juma → j u, u m, m a**
* **wewe → w e, e w, w e**
* **jina → j i, i n, n a**
* **lako → l a, a k, k o**
* **nani → an, a n, n i**
* **maria → m a, ar, r i, i a**
* **nimefurahi → n i, i m, m e, e f, f u, u r, r a, an, h, h i**
* **kukutana → k u, u k, k u, u t, t a, an, n a**
* **na → an**
* **wewe → w e, e w, w e**
* **unaishi → u n, n a, an i, i s, s h, h i**
* **wapi → w a, a p, p i**
* **ninaishi → n i, i n, an a, a i, i s, s h, h i**
* **kisumu → k i, i s, s u, u m, m u**
* **nairobi → n a, a i, i r, r o, o b, b i**
* **unafanya → u n, an f, f a, a n, n y, y a**
* **kazi → k a, a z, z i**
* **daktari → d a, a k, k t, t a, an r, r i**
* **meno → m e, e n, n o**
* **biashara → b i, i a, an s, s h, h a, a r, r a**
* **yangu → y a, a n, ng u**
* **hapa → h a, a p, p a**
* **kabisa → k a, a b, b i, i s, sa**
* **tutaonana → t u, u t, t a, a o, o n, an, n a**

6.

import re

# The ground truth text

ground\_truth\_text = "habari za asubuhi nzuri sana uhali gani salama asante jina langu ni juma na wewe jina lako ni nani jina langu ni maria nimefurahi kukutana na wewe mimi pia nimefurahi kukutana na wewe unaishi wapi ninaishi kisumu na wewe ninaishi nairobi unafanya kazi gani mimi ni daktari wa meno na wewe mimi nina biashara yangu hapa kisumu vizuri kabisa nimefurahi kuwa na rafiki kutoka kisumu asante saso inabidi niende tutaonana baadaye sawa tutaonana baadaye kwaheri kwaheri "

# The pattern matches any word followed by a period then a space which may indicate an abbreviation

abbreviation\_pattern = r'\b[A-Za-z]+\.(?=\s|$)'

# re.sub to remove the abbreviations from the text

cleaned\_text = re.sub(abbreviation\_pattern, '', ground\_truth\_text)

# Print the cleaned text (with abbreviations removed)

print("Cleaned Text:")

print(cleaned\_text)

7.

**Lack of Sufficient Training Data**:

* **Challenge**: One of the primary challenges for low-resource languages is the limited amount of speech data available for training speech recognition systems. Unlike high-resource languages (such as English, Spanish, and Mandarin), low-resource languages lack large, annotated speech corpora. This scarcity makes it difficult to train accurate models that can recognize speech effectively.
* **Recommendation**: To mitigate this, techniques like **data augmentation** (e.g., varying speed, pitch, or noise), **transfer learning** (using models trained on related languages), and **crowdsourcing** (collecting data from native speakers through mobile apps or online platforms) can be explored. Additionally, synthetic speech data can be generated using text-to-speech (TTS) systems to expand the training dataset.

**Dialect Variations and Accents**:

* **Challenge**: Many low-resource languages have diverse dialects or regional variations in pronunciation, which makes it harder to create a generalized speech recognition model. Accents and local pronunciations can significantly affect the performance of speech systems.
* **Recommendation**: Building regional or dialect-specific models and integrating accent adaptation techniques can improve performance. Moreover, multi-dialectal training using data from diverse speakers can help the model generalize better across variations.